Milestone #6

Milestone Overview

You will augment your pipelined MIPSlite microprocessor so that it correctly processes the shift and data manipulation instructions.

This project milestone is worth 40 points. For full credit, your solution must be completed no later than Thursday, March 16. A 10% penalty will be applied for each day that your solution is late (based on the date when your files are submitted via the "handin" system).

Milestone Deliverables

The deliverable for this milestone is:

    machine.c

Be sure to use the specified file name, and to submit your file for grading via the "handin" system.

Milestone Specifications

The MIPSlite microprocessor recognizes 31 machine language instructions. For this milestone, only the instructions in the following subsets are relevant:

    R-format shift instructions: \{SLL, SRL, SRA, SLLV, SRLV, SRAV\}
    R-format data manipulation: \{ADD, SUB, AND, OR, XOR, NOR, SLT\}
    I-format data manipulation: \{ADDI, SLTI, ANDI, ORI, XORI, LUI\}

For this milestone, you will add the Shift Unit and Math Unit to your pipelined machine from Milestone #5 so that it correctly processes the shift and data manipulation instructions, although it need not detect or resolve data hazards.

Create a new project subdirectory under your account and copy the appropriate files, including your "machine.c" and the following instructor-supplied files:

    /user/cse420/Project/Milestone06/machine*

The following instructor-supplied files are available but must not be copied into your account:

    /user/cse420/Project/Milestone06/ClockUnit*
    /user/cse420/Project/Milestone06/FetchUnit*
    /user/cse420/Project/Milestone06/ControlUnit*
    /user/cse420/Project/Milestone06/RegisterUnit*
    /user/cse420/Project/Milestone06/DisplayUnit*
    /user/cse420/Project/Milestone06/ShiftUnit*
    /user/cse420/Project/Milestone06/MathUnit*

Those components behave as specified in the previous milestones.
The following diagram gives the structure of a pipelined implementation of the MIPSlite microprocessor.

The inputs to the Shift Unit are:

- a) the shift unit operation code (2 bits)
- b) the shift amount (5 bits)
- c) the operand to be shifted (32 bits)

For "non-variable" shifts, the shift amount is encoded in the current instruction. For the "variable" shifts, the shift amount is the least significant bits of the register indicated by the "rs" field of the current instruction. A multiplexor will be used to select the correct 5-bit value as the shift amount.

The inputs to the Math Unit are:

- a) the math unit operation code and invert signal (4 bits)
- b) the first operand (32 bits)
- c) the second operand (32 bits)

The second operand will be one of four items; a multiplexor will be used to select the correct alternative:

- port 3: appropriate bit pattern required by "LUI" instructions
- port 2: zero-extended 16-bit immediate value
- port 1: sign-extended 16-bit immediate value
- port 0: contents of register "rt"

Use a multiplexor to route the correct 5-bit register number to the "WriteRegister" input on the Register Unit. Select port 2 when the destination register number is specified by the "rd" field, select port 1 when the destination register number is 31, and select port 0 when the destination register number is specified by the "rt" field.

Use a multiplexor to route the correct 32-bit value to the "WriteData" input on the Register Unit. Select port 2 when the value is the output of the Shift Unit and select port 1 when the value is the output of the Math Unit. Use a temporary signal (with the value 0x01234567) as the input to Port 3 and Port 0 (those two ports will be used for the output of the Branch Unit and Data Memory Unit in subsequent milestones).